EE2703 : Applied Programming Lab
Assignment 8
DFT using Numpy

Sai Shashank GP
EE20B040

May 16, 2022

0.1 Introduction

in this assignment, we are going to see how can we perform DTFT and DFT using FFT and IFFT algorithms built into **numpy**. We will also be using **pylab** for plotting the frequency Spectra that are obtained. At first, we will be looking into some examples and then extend our understanding of concepts to other signals.

0.2 Assignment

0.2.1 Prerequisites

At first, we need to import some libraries and modules which has FFT and IFFT algorithms.

```
import numpy as np
from pylab import *
```

And also, instead of using the signals as a numpy array, we can define a class for each of the signal we are using and assign some attributes to it so that it would be easier for us to plot them and use them. It is done as follows, I'll be showing for the signal $f(x) = \sin 5x$

```
class sin5x:
    def __init__(self, t):
        self.t = t
        self.name = r'$\sin(5x)$'
    def calc(self):
        return np.sin(5*self.t)
```

And I'm also going to define a helper function which helps us to calculate the DFT using *numpy.fft.fft()* and plot the frequency magnitude and phase spectra.

```
def helper_(func, lim=10, start=0, end=2*np.pi, freq=128, plotRed=True):
    '''This function helps us automate the task of solving and plotting'''
    x = np.linspace(start, end, freq+1)[:-1]
    y_=func(x)
    y_name = y_.name
    y = y_.calc()
    Y=fftshift(fft(y))/freq
    w=linspace(-64,64,freq+1)[:-1]
    figure()
    subplot(2,1,1)
    plot(w,abs(Y))
    xlim([-1*lim,lim])
    ylabel(r"$|Y|$",size=16)
    title(rf"Spectrum of {y_name}")
    grid(True)
    subplot(2,1,2)
    plot(w,angle(Y),'ro', visible=plotRed)
    ii=where(abs(Y)>1e-3)
    plot(w[ii],angle(Y[ii]),'go',lw=2)
    xlim([-1*lim,lim])
    ylabel(r"Phase of $Y$",size=16)
    xlabel(r"$k$",size=16)
    grid(True)
    show()
```

0.2.2 Q1

In Q1, we are asked to execute the FFT, shift the x-axis accordingly and plot the DFT of two signals, namely, $\sin 5x$ and $(1 + 0.1\cos(x))\cos(10x)$. We will be having the following code.

```
# Q1 Part a: sin(5t)
def Q1a():
    '''Solves Q1 for sin(5t)'''
    helper_(func=sin5x)

# Q1 Part b: (1+0.1cos(t))cos(10t)
def Q1b():
    '''Solves Q1 for (1+0.1cos(t))cos(10t)'''
    helper_(func=AMexample, lim=15, start=-4*np.pi, end=4*np.pi, freq=512)
```

Fourier Transform of the signal $\sin 5x$ is given to be

$$\boxed{F(\omega) = \frac{1}{2j}(\delta(\omega-5) - \delta(\omega+5))} \tag{1}$$

Fourier Transform of the signal $(1 + 0.1\cos(x))\cos(10x)$ is given to be

$$\boxed{F(\omega) = \frac{1}{2}(\delta(\omega-10) + \delta(\omega+10) + 0.05(\delta(\omega-9) + \delta(\omega+9) + \delta(\omega-11) + \delta(\omega+11)))} \tag{2}$$

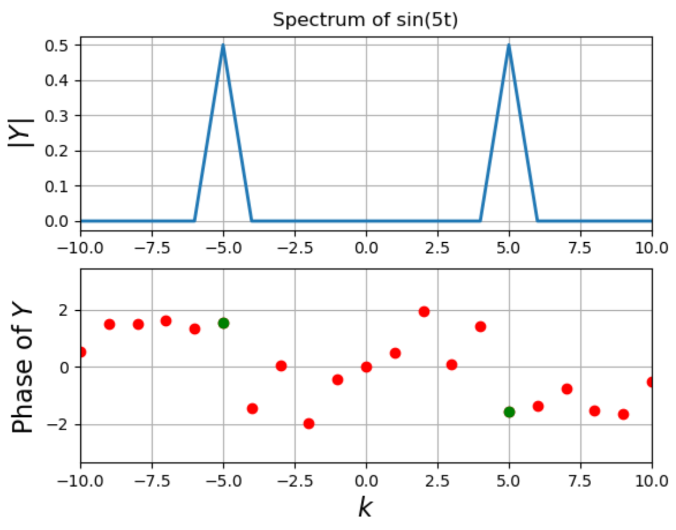

Figure 1: Frequency Spectrum of $\sin 5x$

And we are getting the graphs of the calculated Frequency spectra as expected. I had to sample more for AM signal than given sinusoidal signal.

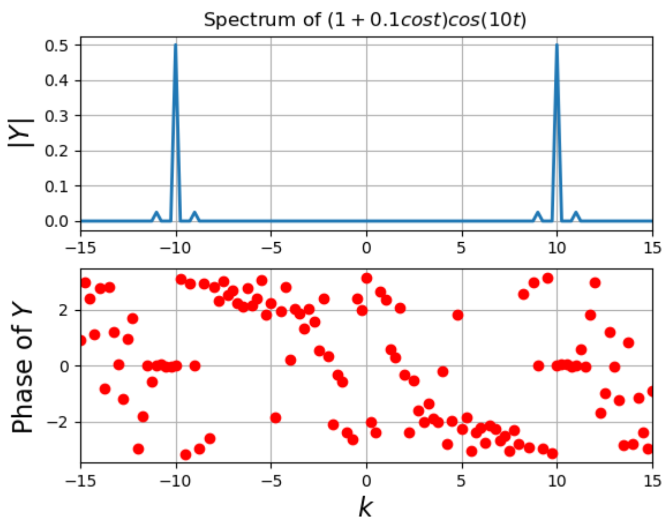

Figure 2: Frequency Spectrum of $(1 + 0.1\cos(x))\cos(10x)$

0.2.3 Q2

In Q2, we are asked to execute FFT, shift the x-axis accordingly and plot the DFT of two signals, namely, $\sin^3 x$ and $\cos^3 x$. We will be having the following code.

```
# Q2 Part a: sin^3(t)
def Q2a():
    '''Solves Q2 for sin^3(t)'''
    helper_(func=sinx3, lim=6, start=-4*np.pi, end=4*np.pi, freq=512)

# Q2 Part b: cos^3(t)
def Q2b():
    '''Solves Q2 for cos^3(t)'''
    helper_(func=cosx3, lim=6, start=-4*np.pi, end=4*np.pi, freq=512)
```

Fourier Transform of the signal $\sin^3 x$ is given to be

$$\boxed{F(\omega) = \frac{1}{4j}(3(\delta(\omega-1) - \delta(\omega+1)) - (\delta(\omega-3) - \delta(\omega+3)))} \tag{3}$$

Fourier transform of the signal $\cos^3 x$ is given to be

$$\boxed{F(\omega) = \frac{1}{4}(3(\delta(\omega-1) + \delta(\omega+1)) + (\delta(\omega-3) + \delta(\omega+3)))} \tag{4}$$

We are getting Frequency Spectra as expected. Sampling is same as AM wave sampling.

0.2.4 Q3

In Q3, we are asked to execute FFT, shift the x-axis accordingly and plot the DFT of an FM wave, given as, $\cos(20t + 5\cos(t))$. We will be using the following code

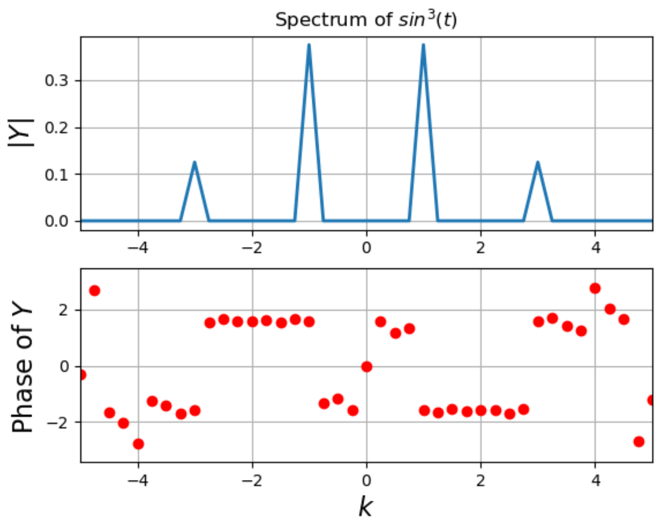

Figure 3: Frequency Spectrum of $\sin^3 x$

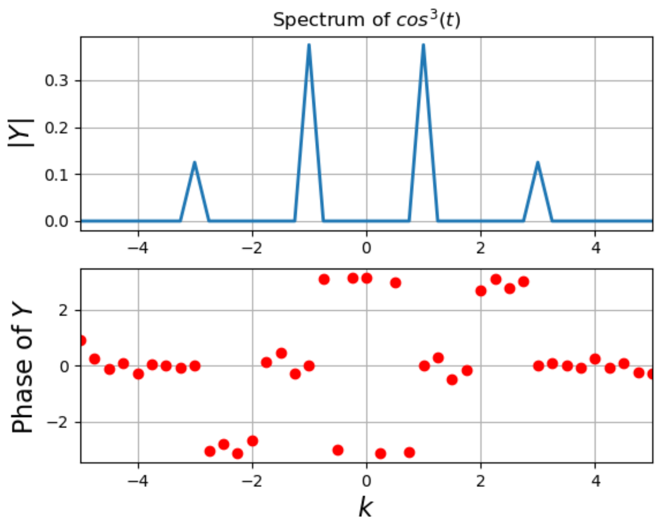

Figure 4: Frequency Spectrum of $\cos^3 x$

```
# Q3: cos(20t+5cos(t)) or example of FM wave
def Q3():
    '''Solves Q3 for cos(20t+5cos(t))'''
    helper_(func=FMwave, lim=15, start=-32*np.pi, end=32*np.pi, freq=8192,
    plotRed=False)
```

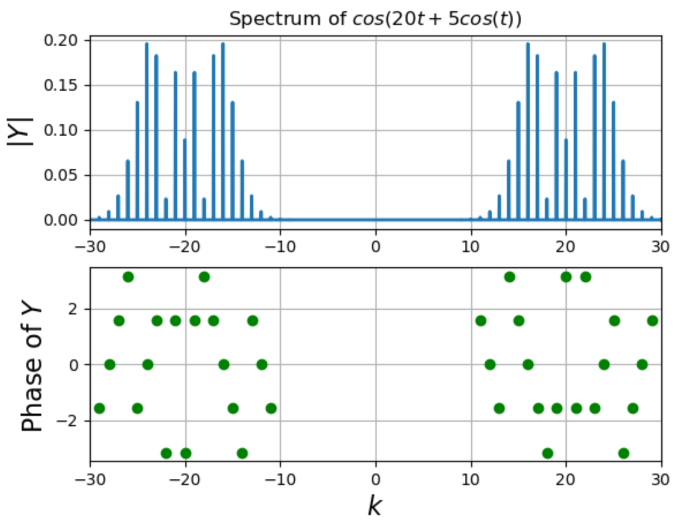

Figure 5: Frequency Spectrum of $\cos(20t + 5\cos(t))$

0.2.5 Observation

As we increase the sampling space, the magnitude plot appears to be changing and diverging from y-axis. It can be observed in the following plots.

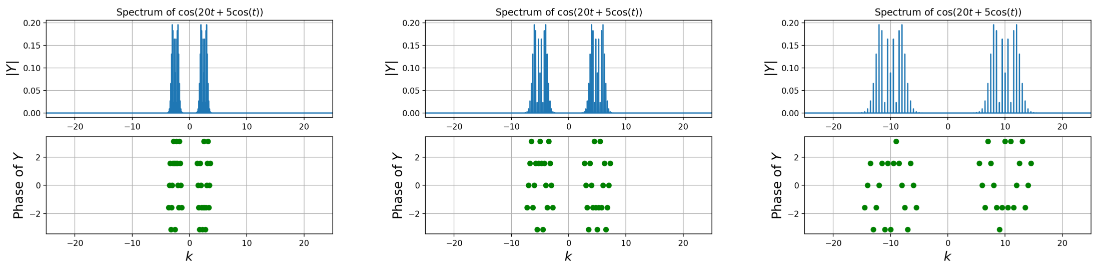

Figure 6: Sampling space increasing two fold as we go to right

0.2.6 Q4

In Q4, we are asked to execute FFT, shift the x-axis accordingly and plot the DFT of a Gaussian signal, given as, $e^{\frac{-x^2}{2}}$. We will be using the following code

```
# Q4: exp(-t^2/2)
def Q4():
    '''Solves Q4 for given Gaussian Signal'''
    helper_(func=Gaussian, lim=10, start=-8*np.pi, end=8*np.pi, freq=1024, plotRed=True)
```

0.2.7 Validation

Now, let's see the plot we got is accurate upto 6 digits or not. So, let's use the following code.

```
def acc_ft(w):
        return np.sqrt(2*np.pi)*np.exp(-0.5*w**2)
```

```
max_err = max(Y - acc_ft(w))
print(max_err)
```

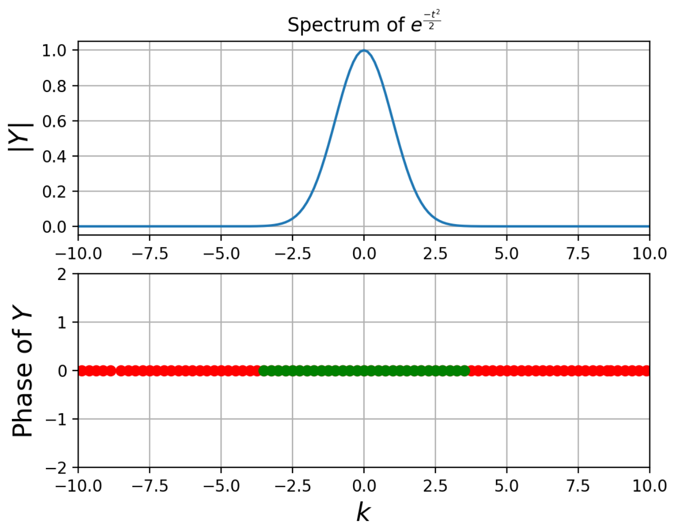

Figure 7: Frequency Spectrum of $e^{\frac{-t^2}{2}}$

It comes out to be **3.86e-16**
So, our plot is valid to the question's requirements

0.3 Takeaways

- I have got some clarity on how important and how sensitive sampling frequency attribute is.
- I have got to know that DFTs can be used to know what frequencies exist in a signal. The peaks in the magnitude spectrum indicate the frequency components of the signal.
- I have got to know that for non harmonic signals like gaussian, which are usually noises, can be identified by sampling more and increasing the threshold to identify frequencies